22

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/802,952	03/12/2001	Junya Yamashita	60188-042	4805

7590 07/12/2004

Jack Q. Lever, Jr.
McDERMOTT, WILL & EMERY
600 Thirteenth Street, N.W.
Washington, DC 20005-3096

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/802,952	YAMASHITA ET AL.
	Examiner	Art Unit
	David J. Huisman	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 May 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 May 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *13 May 2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS and Amendment as received on 5/13/2004.

Specification

3. The title of the invention is not descriptive because the examiner asserts that every processor executes instructions stored in a memory. A new title is required that is clearly indicative of the invention to which the claims are directed. MPEP §606.01 states "This may result in slightly longer titles, but the loss in brevity of title will be more than offset by the gain in its informative value in indexing, classifying, searching, etc. If a satisfactory title is not supplied by the applicant, the examiner may, at the time of allowance, change the title by examiner's amendment."

Maintained Rejections

4. Applicant has failed to overcome the rejections set forth in the previous Office Action for claims 1-20. Consequently, these rejections are respectfully maintained by the examiner and copied below for applicant's convenience.

Maintained Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims **1-3, 5, 7-10,** and **12-15** are rejected under 35 U.S.C. 102(b) as being anticipated by Jaggar, U.S. Patent No. 5,506,976 (as applied in the previous Office Action).

7. **In regard to claim 1:**

8. Jaggar discloses a processor (Fig. 2) comprising:

a) instruction executing means (fig. 2, processing pipeline 2) for executing an instruction stored in storing means (col. 6, lines 44-45, fig. 2, element 8);

b) execution instruction address outputting means (fig. 2, program counter 10 outputs address to be fetched to memory 8) for outputting an execution instruction address that is an address of an area in which an instruction to be executed by the instruction executing means is stored;

c) the processor further comprising:

d) detecting means (fig. 2, program counter 10, reach value register 16, comparator 14, and branch cache 4) for detecting that the instruction to be executed by the instruction executing means is a last instruction of a process before branching (in fig. 1, instruction E is shown to be the last instruction of a process before branching; initially instruction E is detected as the last instruction before branching by detecting the branch instruction by the

branch detection means 22, whereby an entry in the branch cache 4 is created for instruction E [col. 7, lines 39-49], subsequently instruction E is detected by comparing it with the entries in the branch cache [col. 8, lines 15-25] i.e. branch cache hit);

e) wherein the execution instruction address outputting means outputs a start address (&U, fig. 3) that is an address of an area in the storing means in which a first instruction of a process after branching (U, fig. 1) is stored, when the last instruction (E, fig. 1) is detected by the detecting means (col. 8, lines 26-31).

9. In regard to claim 2:

10. Jaggar discloses that the execution instruction address outputting means further comprises of:

a) start address storing means (fig. 2, branch cache 4) for storing a start address (target addresses) of each of a plurality of processes in the storing means;

b) start address selecting means for sequentially switching and selecting the start address stored in the start address storing means, every time the last instruction is detected by the detecting means (when a branch cache hit occurs, a start address [target address] associated with that entry is selected col. 8, lines 31-35),

c) wherein the execution instruction address is output based on the start address selected by the start address selecting means (fig. 4, the program counter 10 receives the start address from the target address latch 18 col. 8, lines 26-31).

11. In regard to claim 3:

12. Jaggar discloses that processor further comprises:

a) end address storing means (fig. 2, branch cache 4) for storing an end address (reach value, col. 6, lines 50-51) that is an address of an area in which a last instruction of each of a plurality of processes is stored in the storing means (in the example given in fig. 3, the address of the last instruction E of a process [see fig. 1] is stored in the reach value of Q' entry);

b) end address selecting means for sequentially switching and selecting the end address stored in the end address storing means, every time the last instruction is detected by the detecting means (when a branch cache hit occurs, an end address [reach value] associated with that entry is selected col. 8, lines 26-28),

c) wherein the detecting means detects the last instruction (col. 6, lines 64-67) based on the execution instruction address output from the execution instruction address outputting means (fig. 2, program counter 10) and the end address selected by the end address selecting means (reach value register 16).

13. In regard to claim 5:

14. Jaggar further discloses that the detecting means detects the last instruction (branch instruction detector 22 detects the last instruction by detecting the branch instruction col. 7, lines 39-49).

15. Although Jaggar does not explicitly mention that the branch instruction detector detects the last instruction based on information stored in correspondence with information indicating a

content of an instruction to be executed by the instruction executing means in the storing means, it is deemed inherent to the branch detection means to detect the branch instruction based on the opcode information of the instruction which is stored in correspondence with the information indicating a content of an instruction to be executed.

16. In regard to claim 7:

17. Jaggar discloses that the start address storing means comprises a memory storing the start address (branch cache 4).

18. Although Jaggar does not explicitly mention that the start address selecting means comprises address designating means for designating an address of an area in which the start address is stored in the memory, It is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

19. In regard to claim 8:

20. Jaggar discloses that the start address stored in the start address storing means can be set by execution of an instruction by the instruction executing means (target address is stored on execution of a branch instruction col. 7, lines 47-52).

21. In regard to claim 9:

22. Jaggar discloses that the start address stored in the start address storing means can be set by a supervisory processor (fig .2, branch instruction detector 22 sets the target address col. 7, lines 39-52) for controlling an operation of the processor.

23. In regard to claim 10:

24. Jaggar discloses start address storing means (fig. 2, branch cache 4) for the supervisory processor (fig. 2, branch instruction detector 22) for storing a start address output from the supervisory processor,

wherein the start address stored in the start address storing means for the supervisory processor is written in the start address storing means at predetermined timing (when the branch instruction's target address is determined, col. 7, lines 48-52).

25. In regard to claim 12:

26. Jaggar discloses that the end address storing means comprises a memory storing the end address (branch cache 4).

27. Although Jaggar does not explicitly mention that the end address selecting means comprises address designating means for designating an address of an area in which the end address is stored in the memory, It is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

28. In regard to claim 13:

29. Jaggar discloses that the end address stored in the end address storing means can be set by execution of an instruction by the instruction executing means (reach value, PC-1, is stored on execution of a branch instruction col. 7, lines 47-49).

30. In regard to claim 14:

31. Jaggar discloses that the end address stored in the end address storing means can be set by a supervisory processor (fig .2, branch instruction detector 22 sets the reach value, PC-1, col. 7, lines 39-48) for controlling an operation of the processor.

32. In regard to claim 15:

33. Jaggar discloses end address storing means (fig. 2, branch cache 4) for the supervisory processor (fig. 2, branch instruction detector 22) for storing an end address output from the supervisory processor,

34. wherein the end address stored in the end address storing means for the supervisory processor is written in the end address storing means at predetermined timing (when the branch instruction is detected, col. 7, lines 39-48).

Maintained Claim Rejections - 35 USC § 103

35. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person

having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

36. Claims **4** and **18-20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar, as applied above, in view of Nair, U.S. Patent No. 6,304,962 (as applied in the previous Office Action, and further in view of Atkins et al., U.S. Patent No. 5,898,866 (as applied in the previous Office Action and herein referred to as Atkins).

37. In regard to claim 4:

38. The processor of Jaggar, which has an end address storing means (fig. 2, branch cache 4) and an end address selecting means (when a branch cache hit occurs, an end address [reach value] associated with that entry is selected col. 8, lines 26-28), differs from the current invention in that it does not have a processing length storing means and a processing length selecting means. Furthermore, while the processor of Jaggar does have an instruction address generating means (fig. 2, program counter 10 outputs address to be fetched to memory 8) and a last instruction detecting means (fig. 2, program counter 10, reach value register 16, comparator 14, and branch cache 4), it does not disclose that the instruction address is generated by adding the start address and a relative address and that the last instruction is detected based on the relative address and the processing length selected.

39. However, Nair teaches of storing the processing length (run length) of a process (superblock) in a superblock target buffer (col. 5, lines 39-46) on the execution of a taken branch instruction (col. 6, lines 64-67; col. 7, lines 1-2). This length is a relative address of an end address to a start address for each plurality of processes (col. 7, lines 1-3). Fetching is continued

until the end address, which is calculated by adding the starting address and the process length (col. 7, lines 6-10).

40. Atkins et al. teach of a relative addressing scheme in which the execution instruction address outputting means adds a starting address (base and index register contents col. 7, lines 63-64) of a process under processing and a relative address (displacement field col. 7, line 64) of the execution instruction address to generate the execution instruction address (effective address col. 7, line 65).

41. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor of Jaggar by replacing the end address (return value in the branch cache 4) storing means with a process length storing means and replacing the end address selection means with a processing length selecting means for sequentially switching and selecting the processing length stored in the processing length storing means, every time the last instruction is detected by the detection means. Furthermore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use relative addressing to generate an execution instruction address by adding the starting address and a relative address and a detecting means which detects the last instruction based on the relative address and the processing length (to calculate the end address) selected by a processing length selecting means.

42. One of ordinary skill in the art at the time of the invention would have been motivated to use the length of a process in place of the end address because it occupies less space and hence translates to savings in hardware. Furthermore one of ordinary skill in the art at the time of the invention would have been motivated to use relative addressing to generate the execution

instruction address because a relative address is smaller than an effective address thus translating in smaller instruction length.

43. In regard to claim 17:

44. Jaggar in view of Nair and Atkins discloses that the processing length storing means comprises a memory storing the processing length (branch cache 4 of Jaggar).

45. Although Jaggar in view of Nair and Atkins does not explicitly mention that the processing length selecting means comprises address designating means for designating an address of an area in which the processing length is stored in the memory, It is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

46. In regard to claim 18:

47. Jaggar in view of Nair and Atkins has taught the processor of claim 4.

48. Nair further teaches that the processing length stored in the processing length storing means can be set by execution of an instruction by the instruction executing means (run length is stored on execution of a taken branch instruction col. 6, lines 64-67; col. 7, lines 1-2).

49. In regard to claim 19:

50. Jaggar in view of Nair and Atkins has taught the processor of claim 4.

51. Nair further teaches that the processing length stored in the processing length storing means can be set on detection of the last instruction (col. 6, lines 64-67; col. 7, lines 1-2).

Although Nair does not explicitly mention that the processing length is set by a supervisory processor, it is deemed inherent to the design that a control logic exists to set the processing length which can include a supervisory processor.

52. In regard to claim 20:

53. Jaggar in view of Nair and Atkins has taught the processor of claim 19.

54. Jaggar in view of Nair and Atkins further teaches processing length storing means (branch cache 4 of Jaggar) for the supervisory processor for storing a processing length output from the supervisory processor,

55. wherein the processing length stored in the processing length storing means for the supervisory processor is written in the processing length storing means at predetermined timing (when the branch instruction is detected, col. 6, lines 64-67; col. 7, lines 1-2 of Nair).

56. Claim **16** is rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar in view of Nair in view of Atkins, as applied above, and further in view of Breeding, "Microprocessor System Design Fundamentals," Prentice Hall, pp.6-7, 1995, (as applied in the previous Office Action and herein referred to as Breeding).

57. **In regard to claim 16:**

58. Although Jaggar in view of Nair and Atkins does not teach that the selecting means comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting a processing length.

59. Jaggar in view of Nair and Atkins differs from the present invention because the processing length storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the processing length.

60. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

61. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the processing lengths in place of a cache memory.

62. One of ordinary skill in the art at the time would have been motivated to use registers for storing the processing lengths because they are a form of high-speed storage, which in turn would result in better performance.

63. Claims **6** and **11** is rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar, as applied above, in view of Breeding, as applied above.

64. In regard to claim 6:

65. Jaggar teaches a start address selecting means for sequentially switching and selecting the start address stored in the start address storing means (when a branch cache hit occurs, a start address [target address] associated with that entry is selected col. 8, lines 31-35)

66. Although Jaggar does not teach that the selecting means comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting a start address.

67. Jaggar differs from the present invention because the start address storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the start address.

68. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

69. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the start addresses in place of a cache memory.

70. One of ordinary skill in the art at the time would have been motivated to use registers for storing the start addresses because they are a form of high-speed storage, which in turn would result in better performance.

71. In regard to claim 11:

72. Jaggar teaches an end address selecting means for sequentially switching and selecting the end address stored in the end address storing means (when a branch cache hit occurs, an end address [reach value] associated with that entry is selected col. 8, lines 26-28)

73. Although Jaggar does not teach that the selecting means comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting an end address.

74. Jaggar differs from the present invention because the end address storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the end address.

75. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

76. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the end addresses in place of a cache memory.

77. One of ordinary skill in the art at the time would have been motivated to use registers for storing the end addresses because they are a form of high-speed storage, which in turn would result in better performance.

Response to Arguments

78. Applicant's arguments filed on May 13, 2004, have been fully considered but they are not persuasive.

79. Applicant argues the novelty/rejection of claim 1 on pages 10-12 of the remarks, in substance that:

> "...Jaggar only discloses that the program counter (PC) stores a value that represents the address of the instruction to be fed to the pipeline. The Jaggar reference does not expressly disclose or suggest that the address it represents is necessarily the start address. As such, the Jaggar reference fails to disclose or suggest that the execution instruction address outputting means outputs a start address, as currently recited by claim 1."
>
> "...the Examiner asserts that the main memory system corresponds to the claimed storing means. However, Jaggar does not disclose or suggest that the first instruction of a process after branching is stored in the main memory system. Rather, main memory system is used merely for transferring instructions into instruction fetch stage."
>
> "...the detecting means of Jaggar does not detect the last instruction of the process, but detects the lowermost 8 bits of the program counter value and reach value. Thus, at a minimum, it is clear that the detecting means of Jaggar does not correspond to the claimed detecting means for detecting that the instruction to be executed by the instruction executing means is a last instruction of a process before branching."

80. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, the examiner asserts that a program counter (PC) is a known component which stores the address of the instruction that is to be fetched from memory. This is shown in Fig.2 where the program counter applies an address to the main memory in order to fetch an instruction from the location corresponding to the address (also see column 6, lines 44-48). Furthermore, as discussed in column 6, lines 8-18, when the last instruction before a branch is detected, the branch's target instruction is placed directly into the pipeline (shown in Table 3). The target instruction is located at a start address (target address) within main memory. And, according to column 8, lines 26-31, this target address is written to the program counter.

b) Regarding the second argument, as discussed in column 6, lines 44-48 and shown in Fig.2, the program counter applies an address to main memory, wherein the instruction located at that address in main memory is fetched. Instructions inherently have to be stored somewhere. Main memory in this case stores instructions which are to be fetched.

c) Regarding the third argument, it is clear from column 6, lines 8-18, and column 8, lines 15-31, that the last instruction before branching is detected in some fashion. Regardless of the fashion, i.e., whether only a portion of the PC is looked at, or whether a reach value is observed, the last instruction is still detected, and this is all that applicant claims (detecting the last instruction).

81. Applicant argues the novelty/rejection of claim 4 on page 13 of the remarks, in substance that:

> "...nowhere in the disclosure of the cited prior art does it disclose or suggest a processing length selecting means for sequentially switching and selecting the processing length stored in the processing length storing means, as currently recited by claim 4. Indeed, the Examiner has not expressly stated what constitutes the corresponding processing length selecting means and how sequentially switching and selecting are accomplished."

82. These arguments are not found persuasive for the following reasons:

a) As discussed in the previous Office Action, Nair has taught storing the processing length. And, this length is used in a calculation which is used to determine how much fetching must be performed. See column 7, lines 6-10. Clearly, if this number is used in a calculation, then it must be selected. This selection would be performed by the processing selection means. In addition, there is not just a single processing length stored for all blocks (processes). Instead, each block has its own respective processing length. Therefore, when dealing with a first block at time X, a first processing length will be selected. However, when dealing with a second block at time Y, a second processing length will be selected. See column 5, lines 39-46. This is sequential switching (i.e., when you switch between blocks, or processes, the length associated with blocks will vary).

Conclusion

83. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
July 7, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100